ประพนธ์ อัศวภาณุวัฒน์

SOFT MAIL

Delphi Technical : โปรแกรมค้นหาข้อมูล WndGREP

แนวความคิดของโปรแกรมนี้ คงเป็นรากฐานที่ดีสำหรับ ผู้ที่ต้องการนำไปพัฒนาใช้งาน อย่างจริงจังต่อไป

ในการพัฒนาโปรแกรมสมัยใหม่ส่วนใหญ่โดยเฉพาะโปรแกรมบนวินโดวส์แล้ว ส่วนใหญ่จะมีการแบ่งโปรแกรมออกเป็นส่วนย่อย ๆ เช่นเป็นยูนิต (Units) หรือเป็นออบเจ็กต์ (Objects) ต่าง ๆ โดยแต่ละยูนิตหรือออบเจ็กต์จะเป็นที่รวมโปรแกรมย่อยและฟังก์ชันพิเศษที่สร้างขึ้นใหม่ทั้งหมดจะถูกเชื่อมโยงเข้าด้วยกันเป็นโปรเจ็กต์ (Project) จำนวนยูนิตที่เชื่อมโยงเข้าด้วยกันจะขึ้นอยู่กับความมุ่งหมาย ความซับซ้อนของงาน และการออกแบบของโปรแกรมเมอร์ ยิ่งถ้าออกแบบโปรแกรมดี ๆ แล้วเราสามารถนำโปรแกรมย่อยที่สร้างขึ้นเป็นยูนิตไปผนวกใช้กับโปรแกรมอื่น ๆ ได้อีก เป็นการประหยัดเวลาในการพัฒนาโปรแกรมอย่างมาก

ปัญหาหลักของผู้ที่สร้างยูนิตไว้มาก ๆ คือ ไม่สามารถจำได้ทั้งหมดว่าโปรแกรมย่อยที่ต้องการใช้อยู่ที่ไหน ยิ่งต้องกลับไปปรับปรุงโปรแกรมที่เคยพัฒนามานานแล้วยิ่งแย่ใหญ่ แทบจะจำไม่ได้เลยด้วยซ้ำ ปัญหานี้หลาย ๆ ท่านที่เป็นแฟนของค่ายบอร์แลนด์คงพอจะเดาได้ว่าเราสามารถใช้โปรแกรมดอสชื่อ GREP.COM ช่วยในการค้นหาได้ เจ้าโปรแกรมตัวนี้เป็นโปรแกรมค้นหาข้อมูลที่น่าทึ่งมาก ทั้งมีขนาดเล็กและค้นหาได้อย่างรวดเร็ว รวมทั้งยังค้นหาแบบหลาย ๆ ไฟล์ได้ด้วย แต่เป็นที่น่าเสียดายที่ไม่มีเวอร์ชันที่เป็นของระบบวินโดวส์แท้ ๆ มาให้เล่น จนมาถึง Delphi เวอร์ชัน 3.0 นี้จึงได้นำมารวมกับส่วนค้นหาข้อมูลในตัวโปรแกรมเอง

ผมเป็นคนหนึ่งที่พบปัญหาเช่นนี้เหมือนกัน จึงได้พยายามพัฒนาโปรแกรมที่มีความสามารถคล้ายกับโปรแกรม GREP.COM (งานนี้สร้างก่อน Delphi 3.0 คลอดครับ) จะว่าไปแล้วผมมีจุดประสงค์แฝงอีกอย่าง ขออนุญาตกล่าวถึงเผื่อจะเป็นแนวทางให้หลาย ๆ ท่านนำไปพัฒนาต่อได้ คือจะนำเอ็นจิน (Search Engine) ของโปรแกรมไปใช้ในระบบงานสารสนเทศที่ผมรับผิดชอบอยู่ งานที่ว่านี้เป็นงานที่น่าสนใจมากครับ คือพยายามปรับระบบงานของบริษัทฯ เราให้เป็นระบบ Office Automation มากที่สุด ตามนโยบายของผู้บริหารของผม งานที่ผมต้องทำคือพยายามลดจำนวนสื่อบันทึกประเภทเอกสารให้มากที่สุด (Paperless) การลดจำนวนไม่ใช่ว่าเลิกพิมพ์เอกสาร หรือย่อให้น้อยลง (เพราะเอกสารบางชนิดมีความจำเป็นที่ต้องเก็บนานมาก อาจจะ 3-5 ปี) แต่เป็นการมองหาสื่อใหม่ ๆ ที่สามารถใช้แทนกันได้ ราคาถูก และที่สำคัญต้องคงทน

ผมเลือกการบันทึกข้อมูลลงบนแผ่น CD-ROM ชนิด Recordable หรือที่เรียกกันทั่วไปว่า "CD-R" ซึ่งก็มีคุณภาพตามที่ต้องการทุกอย่าง ปัญหาต่อไปคือเมื่อบันทึกข้อมูลแล้วจะดึงข้อมูลขึ้นมาใช้ได้อย่างไร ง่ายสุดเลยคือใช้โปรแกรมเท็กซ์อิดิเตอร์ (Text Editor) หรือเวิร์ดโปรเซสเซอร์ (Word Processor) ทั่ว ๆ ไปอ่านขึ้นมา แต่ปัญหาที่ตามมาคือ

1. โปรแกรมบางตัวเช่น NotePad ไม่สามารถอ่านไฟล์ใหญ่มาก ๆ (ขนาด 4-5 MB) ได้ ซึ่งไฟล์ประเภทนี้มีอยู่มากเพราะเป็นไฟล์ Report

2. เครื่องที่นำแผ่น CD-R นี้ไปใช้งานไม่มีโปรแกรมประเภทนี้ติดตั้งอยู่

3. วิธีการใช้ในการค้นหาข้อมูลทำได้ค่อนข้างยาก หรือแตกต่างกันในแต่ละโปรแกรม ผู้ที่ไม่เคยใช้โปรแกรมอาจมีปัญหาการใช้งานเสียเอง

4. รูปแบบแสดงผลไม่ตรงตามความต้องการ

ทางเลือกสุดท้ายที่ผมเลือกคือพัฒนาโปรแกรมขึ้นเอง เพื่อที่จะสามารถปรับแต่งองค์ประกอบให้ตรงตามความต้องการได้ นี่ก็คือที่มาของ Search Engine ที่ผมนำมาใช้ในโปรแกรมค้นหาข้อมูล WndGREP ที่จะกล่าวถึงต่อไปนี้

WndGREP

WndGREP เป็นโปรแกรมค้นหาข้อมูลที่สร้างด้วยคำมาตรฐานของ Delphi และ Windows API ที่มีอยู่ในทุกเวอร์ชัน จึงสามารถใช้ได้บนระบบ Windows 3.x และ Windows 95 มีคุณสมบัติและข้อจำกัดดังนี้คือ

1. สามารถค้นหาคำใน Text ไฟล์เท่านั้น และค้นหาจากต้นไฟล์ไปท้ายไฟล์

2. ค้นหาแบบ Case Sensitive ได้

3. ไม่จำกัดขนาดของไฟล์ที่ต้องการทำการค้นหา

4. ค้นหาคำในไฟล์แบบ ไฟล์เดี่ยว หลายไฟล์ และทุกไฟล์ในไดเรคทอรีได้

5. แสดงผล และทำเครื่องหมายคำที่ค้นพบในไฟล์

Search Engine ที่ง่ายที่สุดในโลก

WndGREP ใช้เทคนิคการค้นหาข้อมูลในไฟล์ โดยการสร้างบัฟเฟอร์ (Buffer) สำรองขนาดเล็ก (32 KB) ขึ้นในหน่วยความจำ เพื่อใช้ในการเก็บข้อมูลชั่วคราว และคัดลอกบล็อกข้อมูลขนาดเดียวกับบัฟเฟอร์บรรจุลงไป พร้อมกับกำหนด Pointer แบบ Null-Terminated String (Pchar) ชี้ไปยังตำแหน่งต้นข้อมูล แล้วจึงค้นหาคำด้วยคำสั่ง StrPos ซึ่งเป็นฟังก์ชันมาตรฐานของ Delphi ด้วยกรรมวิธีนี้เราสามารถค้นหาข้อมูลในกลุ่มแฟ้มข้อมูลขนาดใหญ่ (30-40 MB) ด้วยเวลาเพียงไม่กี่วินาที (ทดสอบบน Windows 95)

สร้างโปรแกรม

โปรแกรม WndGREP จะแบ่งออกเป็น 2 หน้าต่างย่อยคือ หน้าต่างโปรแกรมหลัก และหน้าต่างแสดงผล โดยคอมโปเนนต์ของทั้งสองหน้าต่างแสดงอยู่ในตารางที่ 1 ตัวโปรแกรมหน้าต่างหลักแสดงอยู่ในลิสติงที่ 1

โปรแกรมหน้าต่างแสดงผลแสดงอยู่ในลิสติงที่ 2 และยูนิตเสริมที่ผนวกอยู่ในรูป Global แสดงอยู่ในลิสติงที่ 3

ตารางที่ 1 แสดงรายการคอมโปเนนต์ที่ใช้ในโปรแกรม

Project : WndGrep : หน้าต่างโปรแกรมหลัก : Umain.Dfm			
Fmain : TFMain		Fmain\OpenDialog1 : TOpenDialog	
Caption	WndGrep	Name	OpenDialog1
Height	294	DefaultExt	*.TXT
Name	Fmain	FileName	*.*
Width	600	Filter	Text Files : *.TXT
BorderStyle	bsSingle	.Filter	Any files : *.*
Fmain\Panel1 : Panel1		Fmain\Memo1 : Tmemo	
Name	Panel1	Name	Memo1
Left	4	Left	0
Top	0	Top	119
Height	116	Height	143
Width	587	Width	592
Fmain\BitBtn5 : TBitBtn		Fmain\COBfileList : TComboBox	
Caption	เลือกไฟล์	Name	COBFileList
Left	9	Left	85
Top	11	Top	10
Height	25	Height	24
Width	72	Width	236
Fmain\BitBtn1 : TBitBtn		Fmain\BitBtn2 : TBitBtn	
Caption	ค้นหา	Caption	ยกเลิกการค้นหา
Left	325	Height	410
Top	11	Left	11
Height	25	Top	25
Width	81	Width	81
Fmain\BitBtn3 : TBitBtn		Fmain\Label1 : Tlabel	
Caption	เก็บหน้าต่าง	Caption	Search String
Left	495	Left	14
Top	11	Top	44
Height	25	Height	16
Width	81	Width	84
Fmain\Edit1 : Tedit		Fmain\GroupBox1 : TGroupBox	
Name	Edit1	Caption	Options
Left	100	Left	15

ตารางที่ 1 (ต่อ)

Top	40	Top	65
Height	24	Height	41
Width	476	Width	561
Fmain\CBcs : TCheckBox		Fmain\CBallFiles : TCheckBox	
Checked	false	Checked	false
Left	20	Left	340
Top	20	Top	20
Height	17	Height	17
Width	111	Width	191
Project : WndGrep : หน้าต่างโปรแกรมแสดงผล : UdspForm.Dfm			
FdspForm : TFdspForm		RichEdit1 : TRichEdit	
Caption	Search result	Name	RichEdit1
Left	234	Left	0
Top	125	Top	0
Height	389	Height	357
Width	491	Width	483
Project : WndGrep : Umain.Dfm			
Caption		Align	
Height		Height	
Name		Left	
Width		Top	
		Width	
Align		Caption	
Left		Height	
Name		Left	
Top		Top	
		Width	
Height		Caption	
Left		Height	
Text		Left	
Top		Top	
Width		Width	
Caption		Caption	
Height		Height	
Left		Left	
Top		Top	
Width		Width	

โปรแกรมหลัก Umain.Pas

เป็นหน้าต่างโปรแกรมหลักดังรูปที่ 1 มีกิจกรรมและตัวแปรที่สำคัญอยู่ในส่วน Privated Declarations ดังนี้ คือ

buffer : TbufferP; ใช้เก็บบล็อกข้อมูลที่อ่านไฟล์ สำหรับใช้เป็น Source ในการค้นหาข้อมูล

sourceBuffer : TbufferP; ใช้ทำสำเนาของ buffer ในกรณีการค้นหาข้อมูลแบบ Non-Case Sensitive

str2Search : string; เก็บข้อความที่ต้องการค้นหา

bufferP, str2SearchP, sourceBufferP : pchar; เป็น Pointer ไปยังชุดตัวแปรดังกล่าวข้างต้น

terminateFlg : boolean; เป็น Flag ใช้สำหรับตรวจสอบเมื่อผู้ใช้ต้องการยุติการค้นหา

procedure InitVar; กิจกรรมที่ใช้ในการกำหนดค่าเบื้องต้นให้กับตัวแปรในโปรแกรม

procedure Echo(content : string); โปรแกรมย่อยสำหรับแสดงผลออกหน้าจอ ใช้สำหรับแสดงผลของการทำงานในโปรแกรมย่อยต่าง ๆ

function GetSearchPath(pth : string) : string; ฟังก์ชันกำหนดชื่อไดเรคทอรีที่ต้องการค้นหา

function GetSearchFileExt(fle : string) : string; ฟังก์ชันกำหนดชุดชื่อสกุลที่ต้องการค้นหา ในกรณีที่ค้นหาทั้งไดเรคทอรี

procedure SelectFileOrFolder; กิจกรรมใช้เลือกไฟล์หรือไดเรคทอรีย่อยที่ใช้ในการค้นหา

procedure SearchInFile(fname : string); กิจกรรมอ่านข้อมูลจากไฟล์ fname โดยใช้คำสั่ง BlockRead อ่านข้อมูลขนาดเท่ากับ BUFFERSIZE บรรจุในตัวแปร Buffer ส่งให้กิจกรรม SearchInDataBlock ค้นหาข้อมูล

function SearchInDataBlock(numRead : integer; filename : string) : integer; กิจกรรมค้นหาข้อมูลจะทำการคัดสำเนาจากตัวแปร buffer ไว้ใน sourceBuffer ก่อน เพื่อเก็บเป็นสำเนาในกรณีของการค้นหาแบบ Non-Case Sensitive แล้วแปลงตัวอักษรเป็นตัวอักษรใหญ่ทั้งหมด จึงทำการค้นหาข้อมูลด้วยคำสั่ง p:=StrPos(bufferP, ssSearchP); ถ้าพบข้อมูลจะส่ง Pointer ที่ได้พร้อมกับตำแหน่ง #13#10 ซึ่งเป็นตำแหน่งสุดท้ายในบรรทัดข้อมูลแบบ Text ให้กับกิจกรรม FoundDataAt ทำการประมวลผลต่อไป

Tips

Text ไฟล์ส่วนใหญ่จะบันทึกข้อมูลตัวอักษรรหัสแอสกี 13 (Carriage Return) และรหัสแอสกี 10 (Line Feed) ไว้ท้ายบรรทัดข้อมูลในแต่ละบรรทัด เป็นสัญลักษณ์ของการขึ้นบรรทัดใหม่

ลิสติงที่ 1

```
00001:  // Listing 1.1
00002:  //
00003:  unit UMain;
00004:  // Declare LONGSTRINGS=OFF
00005:  {$H-}
00006:  interface
00007:  uses
00008:    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
00009:    INIFILEs, StdCtrls, Buttons, ExtCtrls, UgrpStru;
00010:  type
00011:    TFmain = class(TForm)
00012:      Panel1: TPanel;
00013:      Label1: TLabel;
00014:      Edit1: TEdit;
00015:      BitBtn1: TBitBtn;
00016:      BitBtn2: TBitBtn;
00017:      BitBtn3: TBitBtn;
00018:      BitBtn5: TBitBtn;
00019:      GroupBox1: TGroupBox;
00020:      CBcs: TCheckBox;
00021:      Memo1: TMemo;
00022:      CBallFiles: TCheckBox;
00023:      OpenDialog1: TOpenDialog;
00024:      COBfileList: TComboBox;
00025:      procedure FormCreate(Sender: TObject);
00026:      procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
00027:      procedure BitBtn5Click(Sender: TObject);
00028:      procedure BitBtn1Click(Sender: TObject);
00029:      procedure Edit1KeyDown(Sender: TObject; var Key: Word;
00030:        Shift: TShiftState);
00031:      procedure BitBtn2Click(Sender: TObject);
00032:      procedure BitBtn3Click(Sender: TObject);
00033:    private
00034:      { Private declarations }
00035:        buffer,sourceBuffer : TbufferP;
00036:        bufferP, str2SearchP, sourceBufferP : pchar;
00037:        fileCount,totalMatchCnt : integer;
00038:        file2Search, path2Search, str2Search : string;
00039:        curFileSize, curSeekPos : longint;
00040:        totalByteCnt : longint;
00041:        terminateFlg : boolean;
00042:        fileFilter : string;
00043:      procedure InitVar;
00044:      procedure ReleaseVar;
00045:      procedure Echo( content : string );
00046:      function GetSearchPath( pth : string ) : string;
00047:      function GetSearchFileExt( fle : string ) : string;
00048:      procedure SelectFileOrFolder;
00049:      procedure SearchInFile( fname : string );
00050:      function  SearchInDataBlock( numRead : integer; filename : string ) : integer;
00051:      procedure FoundDataAt( foundAtP, endAtP : pchar; fileName : string );
00052:      procedure DisplayFoundData( fileName, foundStr : string );
00053:      procedure SearchAllFileInFolder( fpath : string );
00054:    public
00055:      { Public declarations }
00056:    end;
00057:  var
00058:    Fmain: TFmain;
00059:  implementation
00060:  uses UdspForm;
00061:  {$R *.DFM}
00062:  (*-----------------------------------------*)
00063:  procedure TFmain.InitVar;
00064:  begin
00065:    fileCount:=0; file2Search:=''; path2Search:=''; str2Search:='';
00066:    curFileSize:=0; curSeekPos:=0; terminateFlg:=false; totalByteCnt:=0;
00067:    totalMatchCnt:=0; fileFilter:='';
00068:    getmem( buffer, sizeof( TbufferS ) );
00069:    getmem( sourceBuffer, sizeof( TbufferS ) );
00070:    fillchar( buffer^, BUFFERSIZE+1 , #0 ); bufferP:=pchar( buffer );
00071:    fillchar( sourceBuffer^, BUFFERSIZE+1 , #0 ); sourceBufferP:=pchar( sourceBuffer );
00072:    fillchar( str2Search, sizeof( String ),#0 ); str2SearchP:=@str2Search[1];
00073:  end;
00074:
00075:  procedure TFmain.ReleaseVar;
00076:  begin
00077:    freemem( sourceBuffer, sizeof( TbufferS ) );
00078:    freemem( buffer, sizeof( TbufferS ) );
00079:  end;
00080:
00081:  (*-----------------------------------------*)
00082:
00083:  procedure TFmain.Echo( content : string );
00084:  begin
00085:    if ( memo1.lines.count>200 ) then memo1.lines.delete(0);
00086:    memo1.lines.add( content );
00087:  end;
00088:
00089:  function TFmain.GetSearchPath( pth : string ) : string;
00090:  begin
00091:    if ( pos( '\',pth )>0 ) then
```

ลิสติงที่ 1 (ต่อ)

```
00092:      while ( pth[ length( pth ) ]<>'\' ) and ( length( pth )>0 ) do
00093:        delete( pth, length( pth ), 1 );
00094:    result:=pth;
00095:  end;
00096:
00097:  function TFmain.GetSearchFileExt( fle : string ) : string;
00098:  begin
00099:    if ( pos( '.',fle )>0 ) then delete( fle, 1, pos( '.', fle )-1 );
00100:    result:='*'+fle;
00101:  end;
00102:
00103:  procedure TFmain.SelectFileOrFolder;
00104:  begin
00105:    with OpenDialog1 do
00106:    begin
00107:      FileName:='*.*';
00108:      InitialDir:=path2Search;
00109:    end;
00110:    if ( OpenDialog1.Execute ) then
00111:    begin
00112:     COBfileList.enabled:=true; BitBtn1.enabled:=true; BitBtn2.enabled:=true;
00113:    end else
00114:    begin
00115:      file2Search:=''; path2Search:='';
00116:    end;
00117:  end;
00118:
00119:  procedure TFmain.SearchInFile( fname : string );
00120:  var fp : file; numRead,ret : integer;
00121:  begin
00122:    if ( not fileExists( fname ) ) then exit;
00123:    assignFile( fp,fname ); reset( fp,1 );
00124:    curFileSize:=FileSize(fp); curSeekPos:=0; totalMatchCnt:=0; dspLineCount:=0;
00125:    Echo('แฟ้มข้อมูล ['+fname+'] ขนาด : '+IntToStr( curFileSize )+'ไบต์' );
00126:    Repeat
00127:      fillchar( buffer^, BUFFERSIZE+1, #0 ); bufferP:=pchar( buffer );
00128:      blockRead( fp, buffer^, BUFFERSIZE, numRead );
00129:      ret:=SearchInDataBlock( numRead, fname );
00130:      inc( curSeekPos, ret );
00131:      inc( totalByteCnt, ret );
00132:      seek( fp, curSeekPos );
00133:      schedule; if ( terminateFlg ) then break;
00134:    until ( numRead<>BUFFERSIZE ) or ( numRead<1 );
00135:    closeFile( fp );
00136:    Echo('พบข้อมูลจำนวน :'+IntToStr( totalMatchCnt )+'ครั้ง' );
00137:  end;
00138:
00139:  function TFmain.SearchInDataBlock( numRead : integer; filename : string ) : integer;
00140:  var p,pp,mxP,savPP : pchar; ret : integer;
00141:      ssSearchP : pchar; ssSearch : string;
00142:  begin
00143:    SearchInDataBlock:=numRead; ssSearchP:=@ssSearch[1];
00144:    fillchar( ssSearch, sizeof( string ), #0 ); ssSearch:=str2Search;
00145:    sourceBuffer^:=buffer^;
00146:    if ( not CBcs.checked ) then
00147:    begin
00148:      strUpper( bufferP ); strUpper( ssSearchP );
00149:    end;
00150:    mxP:=@buffer[ numRead ]; savPP:=nil;
00151:    repeat
00152:      p:=StrPos( bufferP, ssSearchP );
00153:      if ( p<>nil ) then
00154:      begin
00155:        pp:=StrPos( p,#13#10 );
00156:        if ( pp<>nil ) then
00157:        begin
00158:          FoundDataAt( p, pp, fileName );
00159:          inc( totalMatchCnt );
00160:          bufferP:=pp; savPP:=pp;
00161:        end else
00162:        begin
00163:          if ( savPP<>nil ) then savPP:=savPP+2 else savPP:=p;
00164:          ret:=mxP-savPP; SearchInDataBlock:=numRead-ret;
00165:        end;
00166:      end;
00167:      schedule; if ( terminateFlg ) then break;
00168:    until ( p=nil ) or ( pp=nil );
00169:  end;
00170:
00171:  procedure TFmain.FoundDataAt( foundAtP, endAtP : pchar; fileName : string );
00172:  var s : string; startP : pchar; start,len : integer;
00173:  begin
00174:    startP:=@buffer[1];
00175:    repeat foundAtP:=foundAtP-1; until ( foundAtP=startP ) or ( foundAtP^=#13 );
00176:    foundAtP:=foundAtP+2; len:=(endAtP-foundAtP)+1;
00177:    start:=foundAtP-startP;
00178:    move( sourceBuffer^[ start+1 ], s[1], len ); s[0]:=chr(len);
00179:    DisplayFoundData( fileName,s );
00180:  end;
00181:
00182:  procedure TFmain.DisplayFoundData( fileName,foundStr : string );
```

ลิสติงที่ 1 (ต่อ)

```
00183:  begin
00184:    FdspForm.Show; schedule;
00185:    FdspForm.DisplaySearchResult( fileName,foundStr, str2Search );
00186:  end;
00187:
00188:  procedure TFmain.SearchAllFileInFolder( fpath : string );
00189:  var SearchRec: TSearchRec; ret : integer; rpath,rfile, tmp : ShortString;
00190:  begin
00191:    rpath:=fpath; rfile:='';
00192:    while ( rpath[ length( rpath ) ]<>'\' ) do
00193:    begin
00194:      rfile:=rpath[ length( rpath ) ]+rfile; delete( rpath, length( rpath ), 1 );
00195:    end;
00196:    rpath:=fpath;
00197:    if ( rpath[ length( rpath ) ]<>'\' ) then
00198:       while ( rpath[ length( rpath ) ]<>'\' ) do
00199:         delete( rpath, length( rpath ), 1 );
00200:    ret:=FindFirst( fpath, faAnyFile, SearchRec );
00201:    if ( ret<>0 ) then begin FindClose( SearchRec ); exit; end;
00202:    while ( ret=0 ) do
00203:    begin
00204:      with SearchRec do
00205:      begin
00206:        if ( attr=$00000010 ) then
00207:        begin
00208:          tmp:=name;
00209:          if ( tmp<>'.' ) and ( tmp<>'..' ) then
00210:          begin
00211:            tmp:=rpath+tmp+'\'+rfile;
00212:            SearchAllFileInFolder( tmp );
00213:          end;
00214:        end else
00215:        begin
00216:          if ( attr<>$00000008 ) then
00217:          begin
00218:            if ( pos( '*.*',file2Search )>0 ) then
00219:            begin
00220:              tmp:=rpath+name;
00221:              SearchInFile( tmp );
00222:            end else
00223:            begin
00224:              if ( file2Search[1]='*' ) then delete( file2Search, 1, 1 );
00225:              if ( pos( '.',file2Search )=0 ) then file2Search:='.'+file2Search;
00226:              tmp:=rpath+name;
00227:              if ( pos( Uppercase(file2Search),UpperCase( tmp ) )>0 ) then
00228:              begin
00229:                SearchInFile( tmp );
00230:              end;
00231:            end;
00232:          end;
00233:        end;
00234:      end;
00235:      ret:=FindNext(SearchRec);
00236:    end;
00237:  end;
00238:
00239:  (*-----------------------------------------*)
00240:
00241:  procedure TFmain.FormCreate(Sender: TObject);
00242:  begin
00243:    InitVar;
00244:    with Fmain do begin left:=0; top:=0; end;
00245:  end;
00246:
00247:  procedure TFmain.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
00248:  begin
00249:    ReleaseVar;
00250:    CanClose:=true;
00251:  end;
00252:
00253:  (*-----------------------------------------*)
00254:
00255:  procedure TFmain.BitBtn5Click(Sender: TObject);
00256:  begin
00257:    SelectFileOrFolder;
00258:  end;
00259:
00260:  procedure TFmain.BitBtn1Click(Sender: TObject);
00261:  var c : integer;
00262:  begin
00263:    if ( Edit1.text='' ) then
00264:    begin Echo( 'โปรดระบุข้อมูลที่ต้องการค้นหา...' ); exit; end;
00265:    fillchar( str2Search, sizeof( string ), #0 );
00266:    str2Search:=Edit1.text;
00267:    // Prepare file to search
00268:    if ( CBallFiles.Checked ) then
00269:    begin
00270:      file2Search:=GetSearchFileExt( OpenDialog1.FileName );
00271:      path2Search:=GetSearchPath( OpenDialog1.FileName );
00272:    end else
00273:    begin
```

→

ลิสติงที่ 1 (ต่อ)

```
00274:      fileCount:=OpenDialog1.Files.Count;
00275:      if ( fileCount>1 ) then COBfileList.Items:=OpenDialog1.Files;
00276:     COBfileList.text:=OpenDialog1.FileName;
00277:      file2Search:=COBfileList.text;
00278:      path2Search:=GetSearchPath( OpenDialog1.FileName );
00279:       Echo( 'จำนวนไฟล์ -->'+IntToStr( fileCount ) );
00280:    end;
00281:    Echo( 'ไฟล์ที่ทำการค้นหา -->'+file2Search );
00282:    Echo( 'ค้นใน Folder   -->'+path2Search );
00283:    FdspForm.ClearDisplay;
00284:    // Search file
00285:    if ( CBallFiles.checked ) then
00286:    begin
00287:     SearchAllFileInFolder( path2Search+file2Search );
00288:    end else
00289:    begin
00290:      if ( fileCount=1 ) then
00291:      begin
00292:        file2Search:=COBfileList.Text;
00293:        SearchInFile( file2Search );
00294:      end else
00295:      begin
00296:        for c:= 1 to fileCount do
00297:        begin
00298:          file2Search:=COBfileList.Items[c-1];
00299:          SearchInFile( file2Search );
00300:        end;
00301:      end;
00302:    end;
00303:  end;
00304:
00305:  procedure TFmain.Edit1KeyDown(Sender: TObject; var Key: Word;
00306:    Shift: TShiftState);
00307:  begin
00308:    if ( Key=VK_RETURN ) then
00309:    begin
00310:      if ( not BitBtn1.enabled ) then
00311:      begin Echo( 'โปรดเลือกไฟล์ที่ต้องการค้นหา...' ); exit; end;
00312:      BitBtn1Click( Sender );
00313:    end;
00314:  end;
00315:
00316:  procedure TFmain.BitBtn2Click(Sender: TObject);
00317:  begin
00318:    terminateFlg:=true;
00319:  end;
00320:
00321:  procedure TFmain.BitBtn3Click(Sender: TObject);
00322:  begin
00323:    Fmain.close;
00324:  end;
00325:  end.
```

ลิสติงที่ 2

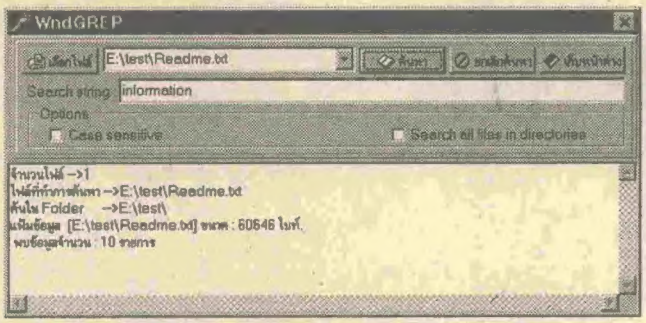
```
00001:  // Listing 1.2
00002:  //
00003:  unit UdspForm;
00004:  interface
00005:  uses
00006:    Windows,. Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
00007:    StdCtrls, ComCtrls, UgrpStru;
00008:  const
00009:    MAX_DISPLAY_LINES = 5000;
00010:  type
00011:    TFdspForm = class(TForm)
00012:      RichEdit1: TRichEdit;
00013:      procedure FormCreate(Sender: TObject);
00014:      procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
00015:    private
00016:      { Private declarations }
00017:        prevFileName : string;
00018:        rStrCount : longint;
00019:      procedure InitVar;
00020:      procedure ReleaseVar;
00021:      procedure Echo( content : string );
00022:    public
00023:      { Public declarations }
00024:      procedure ClearDisplay;
00025:      procedure DisplaySearchResult( fileName, content, searchStr : string );
00026:    end;
00027:  var
00028:    FdspForm: TFdspForm;
00029:  implementation
00030:  uses UMain;
00031:  {$R *.DFM}
00032:  (*-------------------------------------------*)
00033:  procedure TFdspForm.InitVar;
00034:  begin
00035:    prevFileName:=''; rStrCount:=0;
00036:  end;
00037:
```

ลิสติงที่ 2 (ต่อ)

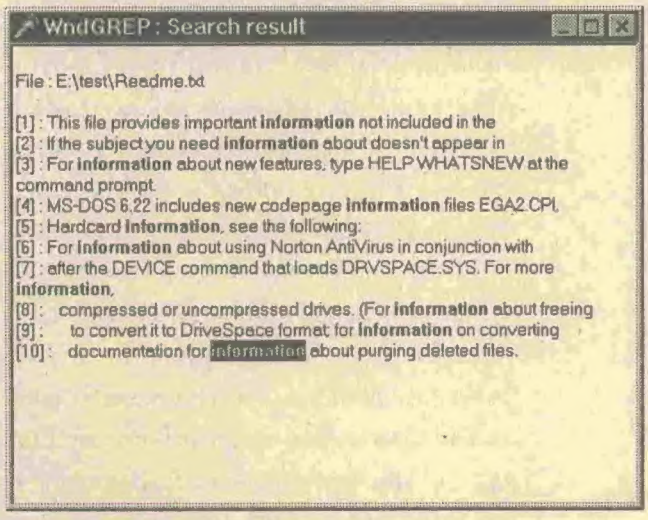
```
00038:  procedure TFdspForm.ReleaseVar;
00039:  begin
00040:    ;
00041:  end;
00042:
00043:  procedure TFdspForm.Echo( content : string );
00044:  var s : string;
00045:  begin
00046:    if ( RichEdit1.lines.count>MAX_DISPLAY_LINES ) then
00047:    begin
00048:      s:=RichEdit1.lines[0]; dec( rStrCount, length( s )+2 );
00049:      RichEdit1.lines.delete(0); schedule;
00050:    end;
00051:    RichEdit1.lines.add( content );
00052:    inc( rStrCount, length( content )+2 ); schedule;
00053:  end;
00054:
00055:  procedure TFdspForm.DisplaySearchResult( fileName, content, searchStr : string );
00056:  var tmp, tmp2 : string;
00057:  begin
00058:    if ( fileName<>prevFileName ) then
00059:    begin
00060:      prevFileName:=fileName;
00061:      Echo( ' ' ); Echo( 'File : '+prevFileName ); Echo( ' ' );
00062:    end;
00063:    inc( dspLineCount );
00064:    Echo( '['+IntToStr( dspLineCount )+'] : '+content );
00065:    with RichEdit1 do
00066:    begin
00067:      if ( not FMain.CBcs.checked ) then
00068:      begin
00069:        tmp:=UpperCase( searchStr ); tmp2:=UpperCase( content );
00070:      end else
00071:      begin
00072:        tmp:=searchStr; tmp2:=content;
00073:      end;
00074:      SelStart:=(rStrCount-(length( content )+2))+pos( tmp, tmp2 )-1;
00075:      SelLength:=length( searchStr );
00076:      with SelAttributes do
00077:      begin Color:=clRed; Style:=[fsBold]; end;
00078:    end;
00079:    schedule;
00080:  end;
00081:
00082:  procedure TFdspForm.ClearDisplay;
00083:  begin
00084:    RichEdit1.lines.Clear; rStrCount:=0; schedule;
00085:  end;
00086:
00087:  (*-----------------------------------------*)
00088:
00089:  procedure TFdspForm.FormCreate(Sender: TObject);
00090:  begin
00091:    InitVar;
00092:    with FdspForm do
00093:    begin Randomize; Left:=( Random( 20 )*10 ); Top:=( Random( 20 )*10 ); end;
00094:  end;
00095:
00096:  procedure TFdspForm.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
00097:  begin
00098:    ReleaseVar;
00099:    canclose:=true;
00100:  end;
00101:  end.
```

ลิสติงที่ 3

```
00001:  // Listing 1.3
00002:  //
00003:  unit UgrpStru;
00004:  interface
00005:  const
00006:    BUFFERSIZE  = 32768;
00007:    SEARCH_DIR  : string[125] = 'C:\';
00008:  type
00009:    TbufferS = array[ 1..BUFFERSIZE+1 ] of char;
00010:    TbufferP = ^TbufferS;
00011:  var
00012:    dspLineCount : integer;
00013:    procedure Schedule;
00014:  implementation
00015:  USES WinTypes;
00016:  procedure Schedule;
00017:  var Msg : TMsg;
00018:  begin
00019:    While PeekMessage(Msg,0,0,0,PM_REMOVE) do
00020:    begin
00021:      TranslateMessage(Msg);
00022:      DispatchMessage(Msg);
00023:    end;
00024:  end;
00025:  end.
```



รูปที่ 1 โปรแกรม WndGREP : หน้าต่างหลัก



รูปที่ 2 หน้าต่างแสดงผลการค้นหา

procedure FoundDataAt(foundAtP, endAtP : pchar; fileName : string); เนื่องจากเราต้องการแสดงผลเป็นบรรทัดแบบ Text กิจกรรมนี้จะทำการค้นหาจุดเริ่มต้นของบรรทัดข้อมูล และแปลงข้อมูลชนิด Pointer ให้เป็น String เพื่อส่งให้กิจกรรม DisplayFoundData แสดงผลต่อไป

procedure DisplayFoundData(fileName, foundStr : string); กิจกรรมนี้จะทำการ Show หน้าต่างแสดงผล และส่งข้อมูลที่ค้นหาพบให้กับหน้าต่างแสดงผล

procedure SearchAllFileInFolder(fpath : string); เป็นกิจกรรมชนิด Recursive จะทำการค้นหาไฟล์ที่อยู่ในไดเรคทอรีทั้งหมด โดยเช็ค attribute ของไฟล์ที่อ่านได้ ถ้ามี attribute เป็น $00000010 หมายถึงไดเรคทอรีจะทำการค้นหาไฟล์ในไดเรคทอรีนั้นด้วยการเรียกตัวเอง และถ้าเป็น $00000008 ซึ่งหมายถึงชนิดไฟล์ ก็จะทำการค้นหาข้อมูลในไฟล์นั้น

โปรแกรมหน้าต่างแสดงผล

เป็นหน้าต่างที่ใช้แสดงผลการค้นหาข้อมูล ดังรูปที่ 2 เนื่องจากต้องการทำเครื่องหมายข้อมูลที่ค้นพบ โดยการเปลี่ยนสีตัวอักษร จึงใช้ TRichEdit เป็นคอมโปเนนต์แสดงผล

กิจกรรมและตัวแปรที่สำคัญในส่วน Private Declarations คือ

rStrCount : longint; เก็บจำนวนข้อมูลในปัจจุบัน ใช้สำหรับคำนวณตำแหน่งเพื่อทำเครื่องหมายสำหรับข้อมูลที่ค้นพบ

procedure InitVar; กิจกรรมกำหนดค่าเบื้องต้นให้กับหน้าต่างแสดงผล

procedure Echo(content : string); กิจกรรมแสดงผลข้อมูลออกหน้าจอ

กิจกรรมและตัวแปรที่สำคัญในส่วน Public Declarations คือ

procedure ClearDisplay; กิจกรรมลบข้อมูลที่แสดงผล เมื่อมีการค้นหาคำใหม่จะเรียกกิจกรรมนี้เพื่อลบข้อมูลเดิม

procedure DisplaySearchResult(fileName, content, searchStr : string); กิจกรรมบรรจุข้อมูลที่ค้นพบให้กับส่วนแสดงผล โดยจะเปลี่ยนสีตัวอักษรที่ค้นพบเป็นสีแดงเพื่อเป็นการเน้นข้อความ

ครับ ทั้งหมดนี้คือโปรแกรม WndGREP จะเห็นว่าตัวโปรแกรมใช้หลักการในการเขียนค่อนข้างง่ายมาก ส่วนที่เขียนยากนั้นน่าจะเป็นส่วนเพิ่มขยาย ซึ่งจะเกี่ยวข้องกับการนำไปประยุกต์การใช้งานเป็นส่วนใหญ่ โปรแกรม WndGREP นี้เป็นเพียงแค่จุดเริ่มต้นของแนวความคิดการสร้างเอ็นจินในการค้นหาข้อมูล ซึ่งการจะนำไปใช้งานจริงคงต้องมีการพัฒนาอีกมาก ผมหวังว่าแนวความคิดของโปรแกรมนี้คงเป็นรากฐานที่ดีสำหรับผู้ที่ต้องการนำไปพัฒนาใช้งานอย่างจริงจังต่อไป สำหรับผู้ที่ไม่ต้องการเสียเวลาในการพิมพ์โปรแกรม WndGREP ผมได้เตรียมโปรแกรมไว้แล้ว เขียนด้วย Delphi 3.0 ท่านสามารถใช้บริการซอฟต์เมล์ได้ สวัสดีครับ